UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/204,973	12/03/1998	DAVID LARS EHNEBUSKE	AT9-98-266	9309

35525 7590 09/24/2003

DUKE W. YEE
CARSTENS, YEE & CAHOON, L.L.P.
P.O. BOX 802334
DALLAS, TX 75380

EXAMINER	
INGBERG, TODD D	
ART UNIT	**PAPER NUMBER**
2124	12

DATE MAILED: 09/24/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS

AND INTERFERENCES

Paper No. 12

Application Number: 09/204,973

Filing Date: December 3, 1998

Appellant(s): EHNEBUSKE ET AL.

Stephan J. Walder, Jr.

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed July 7, 2003.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

Copending application **09/204,971** is also under Appeal. In view of this the Appellent's statement is acknowledged. A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

(3) Status of Claims

The statement of the status of the claims contained in the brief was correct at the time of the filing the Brief. However, due to persuasive arguments additional claims are not deemed allowable at the time of appeal.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Invention

The summary of invention contained in the brief is correct.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

Group I - Claim 1, stands or falls together.

Group II - Claims 12, 13, 15, 23-28, 35, 46, 51, 52, 59, 59, 72, 74-76, 83, 96 and 98, are allowed in view of argument.

Group III - Claim 14, allowed in view of argument.

Group IV - Claims 31, 55 and 79, are allowed in view of argument.

Group V - Claims 32, 43, 45, 56, 67, 69, 80, 91 and 93, are allowed in view of argument.

Group VI - Claims 33, 44, 57, 68, 81 and 92, are allowed in view of argument.

Group VII - Claims 34, 58 and 82,are allowed in view of argument.

Group VIII - Claims 16, stands or falls together.

Group IX - Claims 17 and 22, stand or fall together.

Group X - Claims 18, stands or falls together.

Group XI - Claims 19, stands or falls together.

Group XII - Claims 20, stands or falls together.

Group XIII - Claims 21, stands or falls together.

Group XIV - Claims 23 and 27, are allowed in view of argument.

Group XV - Claims 36-39,41,60-63,65,84-87 and 89, stand or fall together.

Group XVI - Claims 42, 47, 66, 71, 90 and 95, stand or fall together.

Group XVII - Claims 73 and 97 are allowed in view of argument.

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) Prior Art of Record

The following is a listing of the prior art of record relied upon in the rejection of claims under appeal.

James Martin, "Principles of Object-Oriented Analysis and Design", Prentice-Hall International, pages 1 - 412, published June 1, 1992

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C.§ 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless --
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1 - 98 are rejected under 35 U.S.C. 102(a) as being anticipated by "Principles of Object-Oriented Analysis and Design", James **Martin**, published June 1, 1992.

The Martin teaches the underlying theory of building an Object Oriented Computer Aided Software Engineering (OO-CASE) tools in his 1992 text book. The Martin should be taken as a whole, however, focus of the rejection is on the Chapters 9 and 10. The Martin references covers the very basics of object technology that one of ordinary skill should have known well before the time of invention:

Chapter 2 - Basic Concepts

Chapter 3 - Why Object-Oriented ?

Chapter 4 - Basic Guidelines

Chapter 6 - Categorizing Objects

Chapter 7 - Relationships Among Object Types

Chapter 8 - State and State Changes

Chapter 9 - Events, Triggers, and Operations

Chapter 10 - Rules

Chapter 11 - How Diagrams Interrelate

Chapter 12 - Basic Concepts of OO Design

Chapter 15 - Method Creation

Chapter 18 - OO-CASE Tools

Appendix A - Recommended Diagraming Standards

The Martin book in addition to containing foundation knowledge of object oriented technology it teaches applying a set of rules comprising the placing of logic (program statements) in a pre-method control before the logic of a method and post method control point after the logic of a method. Martin also teaches associating a set of rules with each control point based on the class of the object in which the method resides, name of the method and type of control point and invoking methods.

Claim 1

Martin anticipates a computer implemented process for applying a set of rules (**Martin,** Chapter 10, RULES, and page 138-139 and 249-251), the process comprising:

(a) placing a pre-method control point before logic of a method (**Martin,** page 142, operation precondition) and post-method control point after the logic of the method (**Martin,** page 142, postcondition)

(b) associating a set of rules with each control point (**Martin,** page 142, 147 "Operation" as per (a) above) based on a class of object in which the method resides (**Martin,** page 143, "... rules associated with diagrams of OO ...") , name of the method, and type of control point, whether the pre-method control point or the post-method control point (**Martin,** page 142, operation precondition) ;

(c) invoking the method (**Martin**, page 116), wherein encountering each control point during the execution of the method comprises (**Martin,** page 142, postcondition):

(i) determining if the encountered control point is active (**Martin,** page 122, IF structure in center diagram)

(ii) on the basis of an active control point (Interpreted as the result of the IF structure above further described in Appendix A on page 381 Control Conditions):

1) selecting rules based on a set of rules associated with the active control point associated in step (**Martin,** page 122, first diagram example is the control condition to fire missile) (b);

2) running the selected rules (**Martin,** page 122, rule that lead to the control condition);

3) obtaining results from running the rules (**Martin,** page 122, trigger rule at the bottom of the page); and

4) combining the results using a combining algorithm specified by the control point (**Martin,** page 122, A control condition can function as a combining algorithm as seen in diagram in middle of the page and page 126 Figure 9.9 and **Martin** teaches a way to have a combing algorithm where one of three operations are selected as on page 124, and **Martin** teaches a way to have a

combining algorithm where one can be selected as taught in the mutually exclusive notation on the bottom of page 125).

Claim 12

Martin anticipates a computer implemented process for applying a set of rules (as per claim 2), comprising:

(a) invoking a method in an object (as per claim 2);

(b) encountering an active control point during the invocation of the method, wherein the method is a context control point (as per claim 2);

(c) selecting rules associated with the method of the object at the control point (as per claim 2);

(d) invoking the rules (as per claim 2); and

(e) combining results from invoking the rules as per claim 1.

Claim 13

The process of claim 12, wherein the rules perform a variety of actions (Martin, page 164, a variety of actions can occur such as Invoice Student OR Get Dorm depending on the outcome of the Remote Student registered condition) conditioned by the fact that rules may be associated with particular, regularly occurring points in the object model Martin,

(**Martin,** page 166, RULES LINKED TO DIAGRAMS, "The importance of rules was emphasized in Chapter 10 which indicated that rules can be connected to any of the OO diagrams".)

Claim 14

The process of claim 12, wherein the rules perform at least one function which varies over time (**Martin,** page 117, clock events and page 144 Rules Associated with Event Diagrams - " If time is between 9 AM to 5 PM", **Martin**, page 394, Clock Events).

Claim 15

A process of claim 12, wherein a control point occurs just before logic of the method begins, just after the logic of the method completes, or at both just before logic of the method begins and just after the logic of the method completes as per claim 1.

Claim 16

Martin anticipates a computer implemented process for applying a set of rules (as per claim 2) comprising:

(a) defining an object (as per claim 2);

(b) defining at least one method in the object (as per claim 2);

(c) defining at least one control point in the at least one method (as per claim 2).

(d) defining rules to the at least one control point on basis the object's class name, method, name, and position of the at least one control point in the method (Martin, Chapter 12, BASIC CONCEPTS OF OO DESIGN, page 172 - 173, the relationship between classes and objects and the relationship between rules and object modeling).

Claim 17

In the process of claim 16, further comprising the step of activating at least one control point having associated rules as per claim 1.

Claim 18

The process of claim 16 further comprises:

(e) encountering a first control point (**Martin,** page 173, control point with a Time Event) ;

(f) running the rules associated with the first control point (**Martin,** page 173, control point with a Time Event); and

(g) affecting behavior of the object based on running the rules associated with the first control point (The flow control is controlled by the Rule associated with the Control point as per **Martin,** page 381).

Claim 19

In the process of claim 18, the step of affecting the behavior of the object further comprises:

(h) associating different rules to a control point (as per claim 14 - Different rules based on the time of day affects the flow control/ behavior).

Claim 20

In the process of claim 18, the step of affecting the behavior of the object further comprises:

(h) defining another control point (Examiner Interpretation of "defining another control point" the meaning could be at design time or runtime. Design time would involve the interaction with the OO-CASE tool as on **Martin**, page 162, Run time would mean the behavior changes value such as attribute which influence the path the control flow takes. This is the point of programming. The ability to model a problem domain and execute code that process information that reflects the

modeled problem - Flow Control as determined by the running of the program such as **Martin,** page 163).

Claim 21

In the process of claim 18, the step of modifying the object further comprises:

(h) associating rules to a second control point (**Martin,** page 163, Multiple control points defined).

Claim 22

In the process of claim 16, further comprising a step of deactivating the at least one control point.(As per claim 1. The control point is determined if it is active or not. If one takes the **Martin**, page 163 example where the timed event is part of the Waitlisted functionality the Timed event occurs at a specific time the Timed Event is one example of activating and deactivating the control point also see **Martin**, Appendix A, page 394)

Claim 23

Martin anticipates a computer implemented process for applying a set off rules (as per claim 2), comprising

(a) defining an object (as per claim 2);

(b) defining a method in the object (as per claim 2);

(c) defining a first control point of the method, the first control pint being a method context control point (as per claim 2);

(d) determining rules associated with the first control point (as per claim 2);

(e) defining a second control point of the method the second control pint being a method context control point(as per claim 2); and

(f) determining rules associated with the second control point (as per claim 2).

Claim 24

A computer implemented process as in claim 23 further comprising:

(g) separately selecting, running and combining the results of rules determined to be associated with either control point as per claim 1.

Claim 25

In the process of claim 23 wherein the first control point is a pre-method trigger point (**Martin**, page 142, diagram top of page, page 381 Trigger Rules).

Claim 26

In the process of claim 23 wherein the second control point is a post-method trigger point (**Martin**, page 115, Postconditions in cause and effect isolation, page 141, Post Condition page 381, Trigger Rule).

Claim 27

Martin anticipates a computer implemented process for defining an object (Martin, page 166 - 167, Link between, Diagrams, Rules and Objects) comprising:

defining an object; (**Martin**, page 144, Box 10.3, and page 169 - 176 and as per claim 2)

defining a method in the object by: defining method logic (as per claim 2) ;

placing the method logic in the method (**Martin**, page 173, methods is the Specification of an operation and as per claim 2);

defining at least one control point wherein the at least one control point is a method context control point(as per claim 2);

and placing the at least one control point in the method wherein the method logic is continuous wherein the step of placing the at least one control point further comprises placing the at least one control in the method after the method logic. (**Martin**, page 224, DO and FOR loops, page 225, Loops in action diagrams).

Claim 28

A computer implemented process for defining an object as in claim 27, wherein the step of placing the at least one control point further comprises placing the at least one control in the method before the method logic (as per claim 1).

Claim 31

A computer implemented process for defining an object as in claim 27, further comprises: flagging the at least one control point on the basis of being active (as per claim 1).

Claim 32

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprising: defining a rule selection algorithm associated with the at least one control point (**Martin,** page 168, control point rule illustrated).

Claim 33

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprising: defining a rule result combination algorithm associated with the at least one control point. As per claim 1.

Claim 34

A computer implemented process for defining an object as in claim 27, wherein the step of defining the at least one control point further comprises: defining a rule selection algorithm for the at least one control point; and defining a rule result combination algorithm for the at least one control point As per claim 1.

Claim 35

A computer implemented process for defining an object as in claim 27, further comprising: associating at least one rule with the at least one control point. As per claim 32.

Claim 36

Martin anticipates a computer implemented process for defining a rule comprising: creating the rule (**Martin**, page 167, Rule Editor) ; associating the rule with an object class (**Martin**, page 167, Figure 11.14); associating the rule with a method within the object class (Martin, page 173, operations are methods); and associating the rule with an occurrence of a control point within the method and associating the rule with another method within the object class. (**Martin,** page 168, Figure 11.16).

Claim 37

A computer implemented process for defining a rule as in claim 36 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

Claim 38

A computer implemented process for defining a rule as in claim 36 wherein the occurrence of control point within the method being after method logic. As per claim 1.

Claim 39

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another object class (**Martin,** page 267, the ability to access a method/Rule from more than one object and the concept of Reuse which is a key factor in object oriented technology **Martin**, page 248, Box 16.2 Maximize reusability) (This claim could also be interpreted as claiming the principle of inheritance as described on **Martin**, page 266 - 268).

Claim 41

A computer implemented process for defining a rule as in claim 36, further comprising: associating the rule with another control point within the method of the object class (**Martin,** page 166 - 168, the rule associated to the control point , page 233, RULES)

Claim 42

Martin anticipates a computer implemented process for applying a set of rules (as per claim 2), comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules associated with the method comprising: encountering a control point associated with the method; determining if the control point is active; and finding at least one rule

associated with an active control point. (Interpreted as the running of the code generated by claim 2).

Claim 43

A computer implemented process for applying a set of rules as in claim 42, wherein the step of finding at least one rule further comprises: accessing a selecting algorithm associated with the active control point (as per claim 1); and selecting at least one rule using the selecting algorithm (as per claim 10 and The IF structure in the control point as per **Martin**, page 168).

Claim 44

A computer implemented process for applying a set of rules as in claim 42, where in the step of processing rules further comprises: running the at least one rule; determining results from running the at least one rule; accessing a combining algorithm associated with the control point; and combining the results using the combining algorithm. As per claim 1.

Claim 45

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a control point; accessing a selecting algorithm associated with the control point; and selecting at least one rule using the selecting algorithm. As per claim 42.

Claim 46

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a method context control point; finding at least one rule associated with the method context control point; running the at least one rule; determining results on the basis of running the at least one rule; accessing a combining algorithm associated with the method context control point; and combining the results using the combining algorithm. As per claim 1 - the running of the executable generated from the model.

Claim 47

Martin anticipates a computer implemented process for applying a set of rules, comprising: selecting an object class; selecting a method within the object class; invoking the method; processing rules comprising: encountering a first control point associated with the method; determining if the first control point is active (the running of code from claims 1 and 2 and implementations such as page 164 Fig 11.10); executing method logic of the method; encountering a second control point associated with the method; determining if the second control point is active; finding a set of rules associated with one of the first control point and the second control point, wherein the set of rules contains not less than zero rules as per claim 9.

Claim 51

Martin anticipates a data processing system for defining an object comprising: defining means for defining an object; defining means for defining a method in the object by: defining means for

defining method logic; placing means for placing the method logic in the method; defining means for defining at least one control point; and placing means for placing the at least one control point wherein the at least one control point is a method context control point; in the method wherein the method logic is continuous wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method after the method logic.. As per claim 27.

Claim 52

A data processing system for defining an object as in claim 51, wherein the step of placing the at least one control point further comprises placing means for placing the at least one control in the method before the method logic. As per claim 1.

Claim 55

A data processing system for defining an object as in claim 51, further comprises: flagging means for flagging the at least one control point on the basis of being active. As per claim 31.

Claim 56

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point further comprising: defining means for defining a rule selection algorithm associated with the at least one control point. As per claim 32.

Claim 57

A data processing system for defining an object as in claim 51, wherein the step of defining the at

least one control point further comprising: defining means for defining a rule result combination algorithm associated with the at least one control point as per claim 10.

Claim 58

A data processing system for defining an object as in claim 51, wherein the step of defining the at least one control point (as per claims 1 and 2) further comprises: defining means for defining a rule selection algorithm for the at least one control point; and defining a rule result combination algorithm for the at least one control point. As per claim 34.

Claim 59

A data processing system for defining an object as in claim 51, further comprising: associating means for associating at least one rule with the at least one control point. As per claim 8.

Claim 60

Martin anticipates a data processing system for defining a rule comprising: creating means for creating the rule; associating means for associating the rule with an object class; associating means for associating the rule with a method within the object class; associating means for associating the rule with an occurrence of a control point within the method; and associating means for associating the rule with another method within the object class.As per cliam 36.

Claim 61

A data processing system for defining a rule as in claim 60 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

Claim 62

A data processing system for defining a rule as in claim 60 wherein the occurrence of control point within the method being after method logic. As per claim 1.

Claim 63

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another object class. (**Martin,** page 267, the ability to access a method/Rule from more than one object and the concept of Reuse which is a key factor in object oriented technology **Martin**, page 248, Box 16.2 Maximize reusability) (This claim could also be interpreted as claiming the principle of inheritance as described on **Martin**, page 266 - 268).

Claim 65

A data processing system for defining a rule as in claim 60, further comprising: associating means for associating the rule with another control point within the method of the object class. As per claim 1.

Claim 66

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules associated with the method comprising: encountering means for encountering a control point associated with the method; determining means for determining if the control point is active; and finding means for finding at least one rule associated with an active control point. As per claim 42.

Claim 67

A data processing system for applying a set of rules as in claim 66, wherein the step of finding at least one rule further comprises: accessing means for accessing a selecting algorithm associated with the active control point; and selecting means for selecting at least one rule using the selecting algorithm. As per claim 43.

Claim 68

A data processing system for applying a set of rules as in claim 66, where in the step of processing rules further comprises: running means for running the at least one rule; determining means for determining results from running the at least one rule; accessing means for accessing a combining algorithm associated with the control point; and combining means for combining the results using the combining algorithm. As per claim 44.

Claim 69

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point; accessing means for accessing a selecting algorithm associated with the control point; and selecting means for selecting at least one rule using the selecting algorithm. As per claim 45.

Claim 71

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class;

invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a first control point associated with the method; determining means for determining if the first control point is active (as per claim 2); executing means for executing method logic of the method (as per claim 2); encountering means for encountering a second control point associated with the method; determining means for determining if the second control point is active; finding, means for finding a set of rules associated with one of the first control point and the second control point (as per claim 7), wherein the set of rules contains not less than zero rules. As per claim 9.

Claim 72

Martin anticipates a data processing system for applying a set of rules, comprising: selecting means for selecting an object class; selecting means for selecting a method within the object class; invoking means for invoking the method; processing means for processing rules comprising: encountering means for encountering a control point associated with the method; finding means for finding at least one rule associated with the control point prior to executing method logic of the method; running means for running the at least one rule; obtaining means for obtaining results on the basis of running the at least one rule; and controlling means for controlling the method on the basis of the results. As per claim 48.

Claim 73

A data processing system for applying a set of rules as in claim 72, wherein the step of controlling the method comprises: exiting means for exiting the method. As per claim 49.

Claim 74

A data processing system for applying a set of rules as in claim 72, wherein the step of controlling the method comprises: executing means for executing method logic of the method. As per claim 50.

Claim 75

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for defining an object, the instruction comprising: instructions for defining an object; instructions for defining a method in the object by: instructions for defining method logic; instructions for placing the method logic in the method; instructions for defining at least one control point; and instructions for placing the at least one control point in the method wherein the method logic is continuous. As per claim 51.

Claim 76

A computer program product for defining an object as in claim 75, wherein the instruction of placing the at least one control point further comprises placing the at least one control point in the method before the method logic. As per claim 1.

Claim 79

A computer program product for defining an object as in claim 75, further comprises: instructions for flagging the at least one control point on the basis of being active. As per claim 31.

Claim 80

A computer program product for defining an object as in claim 75, wherein the instruction of defining the at least one control point further comprising: instructions for defining a rule selection algorithm associated with the at least one control point. As per claim 32.

Claim 81

A computer product for defining an object as in claim 75, wherein the instruction of defining the at least one control point further comprises: instructions for defining a rule combination algorithm associated with the at least one control point. As per claim 33.

Claim 82

A computer program product for defining an object as in claim 75, wherein the step of defining the at least one control point further comprises: instructions for defining a rule selection algorithm for the at least one control point; and instructions for defining a rule result combination algorithm for the at least one control point.As per claim 34.

Claim 83

A computer program product for defining an object as in claim 75, further comprising: instructions for associating at least one rule with the at least one control point. As per claim 35.

Claim 84

Martin anticipates a computer program product embodied on a computer" readable medium containing instructions for a computer implemented process for defining a rule, the instruction comprising: instructions for creating the rule; instructions for associating the rule with an object class; instructions for associating the rule with a method within the object class; and instructions

for associating the rule with an occurrence of a control point within the method and instructions for associating the rule with another method within the object class.. As per claim 36.

Claim 85

A computer program product for defining a rule as in claim 84 wherein the occurrence of the control point within the method being before method logic. As per claim 1.

Claim 86

A computer program product for defining a rule as in claim 84 wherein the occurrence of control point within the method being after method logic.As per claim 1.

Claim 87

A computer program product for defining a rule as in claim 84, further comprising: instructions for associating the rule with another object class. As per claim 39 or 63.

Claim 89

A computer implemented process for defining a rule as in claim 84, further comprising: instructions for associating the rule with another control point within the method of the object class. As per claim 65.

Claim 90

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an \object class; instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing

rules associated with the method comprising: instructions for encountering a control point associated with the method; instructions for determining if the control point is active; and instructions for finding at least one rule associated with an active control point. As per claim 1.

Claim 91

A computer program product for applying a set of rules as in claim 90, wherein the step of finding at least one rule further comprises: instructions for accessing a selecting algorithm associated with the active control point; and instructions for selecting at least one rule using the selecting algorithm. As per claim 43.

Claim 92

A computer program product for applying a set of rules as in claim 90, where in the step of processing rules further comprises: instructions for running the at least one rule; instructions for determining results from running the at least one rule; instructions for accessing a combining algorithm associated with the control point; and instructions for combining the results using the combining algorithm. As per claim 1.

Claim 93

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a control point; instructions for accessing a

selecting algorithm associated with the control point; and instructions for selecting at least one rule using the selecting algorithm. As per claim 42.

Claim 95

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class (as per claim 42); instructions for selecting a method within the object class; instructions for invoking the method; instructions for processing rules comprising: instructions for encountering a first control point associated with the method; instructions for determining if the first control point is active; instructions for executing method logic of the method; instructions for encountering a second control point associated with the method; instructions for determining if the second control point is active (Asper claim 47); instructions for finding a set of rules associated with one of the first control point and the second control point, wherein the set of rules contains not less than zero rules. As per claim 9.

Claim 96

Martin anticipates a computer program product embodied on a computer readable medium containing instructions for a computer implemented process for applying a set of rules, the instruction comprising: instructions for selecting an object class; instructions for selecting a method within the object class; instructions for invoking the method; processing rules comprising: instructions for encountering a control point associated with the method, the control point being a method context control point; instructions for finding at least one rule associated with the control

point prior to executing method logic of the method; instructions for running the at least one rule; instructions for obtaining results on the basis of running the at least one rule; and instructions for controlling the method on the bass of the results.As per claim 48.

Claim 97

A computer program product for applying a set of rules as in claim 96, wherein the step of controlling the method comprises: instructions for exiting the method. **Martin**, page 236, use of "return" in C++ and it is well known in C++ that reaching the end of a method such as flow control reaching the last "}" in the method declassify will return flow control to the method that called this method or terminate. In either path the method has performed an exit.

Claim 98

A computer program product for applying a set of rules as in claim 96, wherein the step of , controlling the method comprises: instructions for executing method logic of the method. As per claims 50.

(11) Response to Argument

I. Examiner's Interpretation

The Appellant takes issue with the interpretations of the Examiner.

A. The actual interpretations from FAOM and maintained

a. ***Modeling*** - The Martin book teaches the use of modeling an enterprise (**Martin**, page 247 - 249) operation and provides Appendix A, Recommended Diagraming Standards. Martin page 285 illustrates the transition from the problem domain, to modeling, to OO Design to code. The

Martin reference has many chapters covering modeling and discusses the models are tied together to generate code (**Martin,** page 155, box).

The Martin reference also provides some examples. These models are not viewed as static. The Martin reference teaches modeling the enterprise. The principles and techniques are dynamic.

b. ***Flow Control*** - The Applicant does not explicitly claim flow control. However, when the Applicant states in the claims "method logic is continuous" this is interpreted as meaning the method (a common feature in object technology) can run until a outside interrupt occurs. This is a product of flow control resulting from the logic structure of a computer program. There are many claims to what the Examiner interprets as claims to flow control. Flow control is the tracing the path of an executing of a program. The exact path the execution of a program will follow is determined by the values of the attributes and the control conditions encountered. The examples of the programming constructs such as Martin page 148 show the difference paths flow control can take depending on execution of operations such as CHECK IN COPY , FILL REQUEST and BOOK OVERDUE. The values of attributes are tested to determine the path taken. Many claims have made claim to flow control which is inherent to the execution of a computer programs.

c. Claim 3 contains the following limitation ".... the step of defining a first control point further comprises:

(a1) decorating the object to dynamically insert a first control point such that the object acquires this new control point."

The Examiner interprets the "decorating" to mean the entry of programming information such as the operations/methods and the entry of control points in a programming environment.

B. Appellant's Statements

"It is first necessary to address the alleged "interpretations" made by the Examiner of the terms "modeling" and "Flow Control." The Examiner's "interpretations" of these terms in the "Examiner's Interpretation" section of the Final Office Action in no way limits Applicants' claimed invention. The terms in the claims must be interpreted in light of the specification as one of ordinary skill in the art would interpret these terms, not the use of such terms in the reference the Examiner intends to use to reject the claims nor the Examiner's own personal belief as to what the term means.

To the contrary, the Examiner in supposedly "interpreting" the term "modeling" only refers to sections of the Martin reference, discussed hereafter with regard to the rejection under 35 U.S.C. ° 102. The Examiner does not interpret the term "modeling" in light of the present specification. Thus, the claims are not bound by the Examiner's alleged "interpretation." Moreover, the Examiner has merely made a general allegation as to what the term "modeling" is believed to be in view of the Martin reference and has not established in any way how such an "interpretation" has any bearing on any particular feature of any of the rejected claims. Thus, the Examiner's "interpretation" should not be regarded as limiting the scope of the present claims to what the Martin reference teaches, but rather to what the claimed features recite in view of the present specification.

Furthermore, with regard to the "interpretation" of the term "Flow Control", the Examiner admits that Applicants do not claim flow control. Thus, the claims cannot be limited by the Examiner's "interpretation" of the term "Flow Control." With regard to whether what is recited in the claims may be "a product" of "flow control," this is irrelevant to whether the claims as a whole are directed to patentable subject matter. Flow control is not claimed and thus, the claims should not be limited to the Examiner's interpretation of the meaning of "flow control". The mere use of the phrase "method logic is continuous" in any of the claims does not suddenly invoke an interpretation of "flow control" such as that alleged by the Examiner in any manner that would limit the scope of those claims.

With regard to the Examiner's "interpretation" of the term "decorating," the present specification on page 4 clearly describes a mechanism by which a decorator pattern is used to add a new behavior to an object to thereby generate a "decorated" object. Thus, the term "decorating" in claim 3 should be interpreted in light of the specification, not the personal interpretations of the Examiner.

These arguments were first presented to the Examiner in Appellants' Response filed January 30, 2002. In response to Appellants' arguments, the Examiner in the Final Office Action alleges that these terms are terms used by artisans of ordinary skill in the art. With regard to the term "modeling" the Examiner alleges that Appellants' are unable to "distinguish the term from the specification." It is not necessary for Appellants to "distinguish the term from the specification" since the Examiner has not shown how the interpretation of the term "modeling" has any bearing

on the claimed invention. The only place that any form of the term "modeling" is used in the rejected claims is in claim 13 which recites "wherein the rules perform a variety of actions conditioned by the fact that rules may be associated with particular, regularly occurring points in the object model" (emphasis added). Therefore, the interpretation of the term "modeling" is irrelevant to all of the other claims since none of these claims recite "modeling." These other claims include claim 12 from which claim 13 depends and thus, claim 12 contains features that are not limited by the "interpretation" of the term "modeling" alleged by the Examiner.

With regard to the term "flow control" the Examiner responds to Appellants' arguments by stating that Appellants are matching terms and not concepts. Appellants are not merely matching terms. To the contrary, Appellants are alleging that "flow control," as the Examiner interprets this term, has nothing to do with the claimed invention and thus, the Examiner is alleging interpretations of terms that have no bearing on the scope of the claims. Appellants are only requesting that the claims be interpreted based on the features recited therein in view of the specification - not some other unclaimed and unrelated term and "concept" the Examiner wish to "interpret" the claims in view of. However, the Examiner continues to insist on attempting to bring in unclaimed "concepts" in an attempt to try and provide an appearance that the cited reference, Martin, teaches more than it actually does as well as read in limitations to the claims that are not there. Appellants' request that the Board clearly state the improper nature of the Examiner's actions in this regard and that the present claims are not to be limited by the Examiner's irrelevant interpretations of terms and "concepts" discussed above.

In response to Appellants' arguments with regard to the term "decorating", the Examiner states that Appellants have been "provided another opportunity to make a distinction but apparently is unable to do so." In Appellants' response to the Examiner's interpretation of the term "decorating" Appellants pointed to page 4 of the specification as teaching a mechanism for decorating an object and asserted that this term in the claims should be interpreted in light of the specification. Thus, Appellants have offered a source for interpreting this term in Appellants' own disclosure, however the Examiner continues to try and insert limitations into the claims by "interpreting" this term in his own way without regard to what Appellants' own disclosure describes. This is clearly improper and Appellants again request that the Board indicate that the Examiner's interpretation of the term "decorating" not be allowed to be limiting on the present claims.

In summary, none of the Examiner's personal interpretation or alleged interpretations in view of the Martin reference with regard to the above terms may be used to limit the scope of the pending claims for the reasons noted above. To the contrary, the terms in the claims should be interpreted in light of Appellants' specification and not the references cited against the application or the Examiner's own personal beliefs.

C. Examiner's Disposition on Appellant's Arguments on Interpretations

First, the interpretation are not "alleged" they are **actual interpretations** in the record. To believe the interpretations improper the Board would have to believe that the invention and Martin reference do not support the interpretations of the Examiner for the terms "Modeling",

Flow Control" and "Decorating" (i.e. programming) as presented above. The result would mean the following:

Modeling - To believe the invention and reference do not support incorporate **modeing** is to believe programming constructs that model real world problems. The most basic definition applies. Martin teaches modeling in many forms. The cited portion of the reference is a form of modeling program constructs in Martin is the same as Appellent's invention.

Flow Control - To believe the invention and reference do not involve program constructs that influence flow control. Such as conditional statements (rules, IF etc). This term is commonly introduced in a introduction to programming course and was intended to assist in understanding the rejection. This term should have been well known to one of ordinary skill prior to invention. Examiner consitently used a basic term in the art to explain the control flow and the role of program contructs used to model real world problems.

Decorating - Appellant's states the true definition of "decorating" is " a mechanism by which a decorator pattern is used to add a new behavior to an object to thereby generate a "decorated" object.". The Examiner interpreted the term to mean "the entry of programming information such as the operations/methods and the entry of control points in a programming environment." If one of ordinary skill should know a "pattern" by definition is a term of the art. A pattern is a class(es). A class is by definition attributes (data) and methods (operations). Objects are by definition made of classes (pattern(s)). These basic concepts of the enabling technology of object oriented technology is required to be of ordinary skill in the art. The Board to believe the Examiner's

interpretation is wrong would need to be able to distinguish between the programming of objects and decorating of objects.

To believe the Appellant that the interpretations are improper is more limiting to Appellant. For this would exclude the Appellant from asserting their invention supports these basic well known functions that should have been known to one of ordinary skill in the art prior to the time of invention. The Martin reference anticipates Modeling and Decorating in the broadest reasonable interpretation in view of the specification. The term **flow control** was presented merely to assist in understanding the rejection. In basic words the programming constructs of Martin (used to model real world problems). The model based on the conditions at run time determine the flow control. "Allegedly" these basic terms and concepts differ from the Appellant's use of the terms. Visibly absent is Appellant's explaination of how the terms differ. Appellant's argument's of the interpretations is not persuasive.

II. 35 USC § 102, antcipation

A. Group I- independent claim 1 stands or falls on the limitations of claim 1.

Limitations of Claim 1

Claim 1

A computer implemented process for applying a set of rules, the process comprising:

(a) placing a pre-method control point before logic of a method and post-method control point after the logic of the method;

(b) associating a set of rules with each control point based on a class of object in which the method resides, name of the method, and type of control point, whether the premethod control point or the post-method control point;

(c) invoking the method, wherein encountering each control point during the execution of the method comprises:

(1) determining if the encountered control point is active;

(ii) on the basis of an active control point:

1) selecting rules based on a set of rules associated with the active control point associated in step (b);

2) running the selected rules;

3) obtaining results from running the rules; and

4) combining the results using a combining algorithm specified by the control point.

Appellent's Arguements

"With regard to claim 1, the Office Action states:

Martin anticipates a computer implemented process for applying a set of rules (Martin, Chapter 10, RULES, and page 138-139 and 249-251), the process comprising:

(a) placing a pre-method control before logic of a method (Martin, page 142, operation precondition) and post method control point after the logic of the method (Martin, page 142, post condition)

(b) associating a set rules with each control point (Martin, page 142, 147 "Operation" as per (a) above) based on a class of object in which the method resides (Martin, page 143, "...rules associated with diagrams of 00..."), name of the method and type of control point, whether the pre-method control point or the post-method control point (Martin, page 142, operation precondition);

(c) invoking the method (Martin, page 116), wherein encountering each control point during the execution of the method comprises (Martin, page 142, post condition)

(i) determining if the encountered control point is active (Martin, page 122, IF structure in center diagram);

(ii) on the basis of an active control point (Interpreted as the result of the IF structure above further described in Appendix A on page 381 Control Conditions);

1) selecting rules based on a set of rules associated with the active control point associated in step (Martin, page 122, first diagram example is the control condition to fire missiles)(b);

2) running the selected rules (Martin, page 122, rule that lead to the control condition);

3) obtaining results from running the rules (Martin, page 122, trigger rule at the bottom of the page); and

4) combining the results using a combining algorithm specified by the control point (Martin, page 122, A control condition can function as a combining algorithm as seen in diagram in middle of the page and page 126 Figure 9.9 and Martin teaches a way to have a combining algorithm where one of three operations are selected as on page 124, and Martin teaches a way to

have a combining algorithm where on can be selected as taught in the mutually exclusive notation on the bottom of the page 125).

A prior art reference anticipates the claimed invention under 35 U.S.C.§102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. In re Bond, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. In re Lowry, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that Martin does not identically show each and every feature of the pending claims arranged as they are in the claims. Furthermore, Applicants respectfully submit that the claims do not read on the processes specifically described in the Martin reference and that the Office Action is engaged in applying broad "conceptual" teachings of Martin without regard for the actual specific teachings of Martin or the specific features recited in the claims.

Chapters 9 and 10 of Martin, which are the basis for the Office Action's rejection of all of claims 1-98, teach a method of modeling the behavior of an object oriented system. The modeling involves representing operations, preconditions of the operations, post conditions of the operations, control conditions, events and triggers. In the modeling described by Martin, an operation may have a precondition that identifies what must happen before the operation

executes, and a post condition that describes the result of the operation if the operation executes with the precondition being satisfied. The operation may further include a control condition which is a condition that is used to determine whether the operation is to execute or not. The control condition must be checked prior to invoking the operation and may be a complex collection of Boolean conditions (page 122). Martin further teaches the implementation of these preconditions, post conditions and control conditions as rules in Chapter 10.

Martin does not teach "associating a set of rules with each control point haled on a class of oect in which the method resides, name of the method and the control point determining if the encountered control point is active," and "selecting niles haled on a set of rules associated with the active control point associated in step (b)," (emphasis added) as recited in claim 1.

With regard to the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point," the Office Action alleges that this feature is taught by Martin on pages 142, 143, and 147. However, these sections of Martin have nothing to do with associating a set of rules with a control point based on a class of obopct in which the method resides, name of the method and a of control point. Page 142 merely describes that the event diagram is an executable diagram from which program code may be generated using a tool such as the OO-CASE tool. Page 143 merely describes that rules may be either object state rules or object behavior rules; that rules may be used with other types of diagrams other than event diagrams; and that rules may be stated as

English expressions which may then be used to generate code. Page 147 merely shows examples of an operation, event diagram and state transition diagram that have rules attached to them.

There is nothing in these sections, or any other sections, of Martin that can remotely be considered to teach the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and the of control point" (emphasis added). It is not clear where or how the Office Action can extract any teaching from Martin that even suggests the features of the presently claimed invention. Rather, it appears that the Office Action is engaged in taking broad teachings of event diagrams and generating code from event diagrams, and reading into these broad teachings the very specific features recited in Appellants' claims.

However, the Office Action cannot read into the prior art features that are only present in Applicants' own disclosure and use the "modified" prior art to support a rejection of the claims. This modification of the prior art being made by the Examiner in the Final Office Action is more akin to a rejection under 35 U.S.C. § 103(a) than the actual basis of the rejection which is 35 U.S.C. § 102(a). As was established for obviousness rejections, an applicant's teachings may not be read into the prior art. Panduit Corp. v. Denison Mfg. Co., 810 F.2d 1561, 1575 n. 29, 1 U.S.P.Q. 1593, 1602 n. 29 (Fed. Cir. 1987). While the court is addressing an obviousness determination in Panduit, the same principle applies even moreso to a rejection under 35 U.S.C. § 102 rejection since under § 102 the rejection must be based solely on the teachings of the

references themselves. In this case, there is nothing in the Martin reference that can be used to support a position of anticipation with regard to the above feature.

Similarly, there is nothing in the Martin reference that remotely even hints at the feature of "determining if the encountered control point is active." The Martin reference does not even recognize a possibility of having active or inactive control points. The Office Action equates the operation precondition and post conditions to the control points recited in the claims, even though they are not the same as the recited control points. A control point, as defined in the present specification is a point at which rules may be attached to add additional functionality. The preconditions and post conditions of Martin are merely requirements for the operation to execute properly.

However, assuming that the preconditions and post conditions are the same as a control point, arguendo, Martin describes the precondition and post conditions as always having to be satisfied in order for proper operation of the object oriented system model. Thus, the precondition and post conditions must always "active" in the Martin reference and there is no need to determine if they are active or inactive. It is for this reason that Martin does not mention anywhere in the entire reference, any step of determining if a control point is active.

The Office Action alleges that this feature is taught in Martin at page 122 simply because Martin teaches an IF structure. The Office Action equates the precondition with the control point and then states that the presence of an IF structure in the precondition is the same as determining if a control point is active. This does not make any sense since, as is clearly described in the

present invention and recited in the claims, if a control point is not active, the rules of the control point are not executed. If the precondition of Martin were the same as a control point, then the precondition must always be active in order for the IF structure to even operate. Therefore, the IF structure would always indicate that the precondition is active and thus, there is no need for the IF structure. Thus, the IF structure is not the same as the step of determining if a control point is active.

The Examiner is attempting to generalize the claimed invention and generalize the teachings of the reference in an attempt to reject the generalization of the claim as being "anticipated" by the generalization of the reference. Claim 1 clearly recites "determining if the encountered control point is active." There is no such similar step in Martin. The "IF" structure referenced by the Examiner is a precondition that is used to determine if a requisite condition exists for the following operation to be performed. There is no teaching in Martin regarding turning this condition on or off, i.e. making it active or inactive. If the precondition exists, it is required for the operation to be performed.

Just as with the above, Martin also provides no teaching that is remotely similar to the feature of "selecting rules based on a set of rules associated with the active control point associated in step (b)." While Martin teaches that rules may be used to implement the precondition and post condition of an operation, there is no teaching in Martin of associating a set of rules with an active control point and then selecting rules based on this set of rules. Martin provides no teaching at all regarding selection of rules and the Office Action has not pointed out

with particularity any section of Martin that teaches a selection of rules based on a set of rules associated with the active control point.

The Office Action alleges that this feature is taught by Martin in the first diagram on page 122 (although the following text indicates that the Examiner is in actuality referring to the last figure on page 121). These diagrams merely illustrate the use of control conditions that are checked prior to an operation executing. There is no selection of rules even mentioned or shown in these figures, let alone the selection of rules based on a set of rules associated with an active control point. That is, using the middle figure of page 122 as exemplary, there is no teaching in Martin to select the second AND clause from the IF statement in the figure as opposed to selecting the third AND clause. All of the conditions set forth in the IF clause in the figure must be satisfied in order for the operation to execute. There is no selection of rules from a set of rules associated with a control point upon determining that the control point is active during execution of a method, as recited in claim 1. The Office Action is referencing portions of the Martin reference that do not even have anything to do with the features of the claim.

Furthermore, the figures referred to in the Office Action cannot be interpreted in any way to teach the selection of rules based on a set of rules associated with an active control point. The figure illustrating firing of a missile is provided to show how a control point may be used to handle a plurality of triggers that are required for the operation to execute. The first diagram on page 122 also illustrates this concept. However, nowhere in the figures of the accompanying text

is there anything mentioned about selection of rules based on a set of rules associated with an active control point.

On page 122 Martin does state that the control condition may be a collection of "or" conditions applied to triggers (see the middle diagram on page 122). However, the "or" Boolean condition does not constitute a selection of rules based on a set of rules associated with an active control point. Rather, the "or" conditions merely operate to state that if any one of the triggers satisfies the control condition, then the operation will execute. There is no selection of rules.

Thus, Martin does not teach each and every feature of claim 1 as is required under 35 U.S.C. § 102(b). Accordingly, Applicants respectfully request withdrawal of the rejection of claim 1 under 35 U.S.C. § 102(b)."

Examiner's Response to Appellants' Argument on Amendment-, Regarding Group I

In response to Appellants' arguments, with regard to the feature of "associating a set of rules with each control point based on a class of object in which the method resides, name of the method and type of control point," the Examiner states that "the reference explicitly states the diagrams generate code not "might" as the Applicant has stated" (Final Office Action, page 35) and that Appellants' arguments amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references (Final Office Action, page 36). Appellants respectfully disagree.

Appellants have set forth in their arguments the specific claim language that is not taught in the reference. Appellants has stated that the portion of the Martin reference cited by the

Examiner teaches that the event diagram is an executable diagram from which program code may be generated using a tool such as the 00-CASE tool (Page 142), that rules may be either object state rules or object behavior rules; that rules may be used with other types of diagrams other than event diagrams; and that rules may be stated as English expressions which may then be used to generate code (Page 143), and examples of an operation, event diagram and state transition diagram that have rules attached to them (Page 147). Appellants have asserted that there is nothing in these cited sections of the Martin reference that teach a set of rules being associated with each control point "based on a class, of object in which the method resides, name of the method and t;me of control point" (emphasis added). The Martin reference does not teach anything like this feature as set forth in the above arguments. Thus, Appellants' argument clearly satisfy the requirement of 37 CFR 1.111 (b) with regard to "specifically pointing out how the language of the claims patentably distinguishes them from the references", i.e. there is nothing in Martin that is remotely similar to a set of rules being associated with each control point "based on a class of object in which the method resides, name of the method and type of control point."

With regard to the determining if an encountered control point is active, the Examiner states that the arguments do not satisfy 37 CFR 1.111 (b) and that the term "active" is being interpreted as a factor of flow control and that this is inherent in programming and "not able to be separated out." Appellants respectfully submit that Appellants have stated the features of the claims that are not taught by the reference, what the reference actually teaches at the cited

portions, and asserted that these teachings are not sufficient to anticipate the claimed feature. Thus, Appellants have satisfied the requirements of 37 CFR 1.111(b).

In addition, there is nothing in the general concept of "flow control" that teaches to determine if a control point that is encountered during the execution of a method is active. The Examiner alleges in his "interpretation" of "flow control" that flow control is the tracing of the path of an executing of a program and that the exact path the program follows is determined by the values of the attributes and the control conditions encountered (Final Office Action, page 3). Even if what the Examiner alleges (without any support from any cited references) is true, this has no bearing on the specific features recited in claim 1. Even if flow control generally teaches that paths of execution are determined based on values of attributes and control conditions, this in no way anticipates the specific feature of determining if a control point is active when it is encountered during execution of a method, as recited in claim 1. In fact, this alleged general teaching does not even mention control points, that they can be active or inactive, that they are encountered during the execution of a method, or that when they are encountered during execution of a method that they are checked to determine if the encountered control point is active.

Moreover, there is nothing in the cited reference, even assuming that the precondition and post-conditions of Martin are equivalent to a pre-method control point and a post-method control point, which they are not, that teaches checking to see if the precondition is active or not. Again, the Examiner is generalizing the alleged teaching of "flow control", generalizing the specific

features of the claimed invention, and then alleging that the claimed invention is anticipated because the two generalization are equivalent. This is clearly improper and does not meet the requirements of establishing anticipation under 35 U.S.C. § 102.

Additionally, the Examiner states that this feature is anticipated because "the Martin reference teaches a plurality of programming constructs," alleges that Appellants are limiting the use of the constructs and alleges that the reference does not put limits on the constructs. In other words, the Examiner is admitting that the Martin reference only teaches generalities while the present claims are reciting specific features. Appellants respectfully submit that this is the very problem with the Examiner's rejection of the claims, i.e. the Examiner is not examining the specific features recited in the claims but has generalize both the claimed invention and the Martin reference to a point where he believes he can allege that the reference anticipates the claimed invention. This is clearly improper because the Examiner is not taking into account each and every feature of the invention as recited in the claims. Moreover, this statement is practically an admission by the Examiner that he has not shown where each and every feature of the claimed invention is "identically shown" in the Martin reference "arranged as they are in the claimed invention" as is required under 35 U.S.C. § 102 but instead has shown alleged constructs without limitations.

With regard to Appellants' argument regarding the feature of "selecting rules based on a set of rules associated with the active control point associated in step (b)," the Examiner responds by stating that the disagreement appears to be based on the term "active" which the Examiner

interprets as a factor of flow control. The issue with regard to "flow control" has been address above. Moreover, the "disagreement" is not based on the term "active" but is rather based on the fact that the Martin reference does not teach a set of rules being associated each control point based on a class of object, a name of the method, and a type of control point, does not teach determining if an encountered control point is active, and thus, does not teach selecting rules based on the set of rules associated with the active control point on the basis of the control point being determined to be active. Not so much as one of the above features have been shown to be present in the Martin reference, whether or not the term "active" can be interpreted as being part of "flow control." Thus, the Examiner has not established anticipation with regard to claim 1.

In view of the above, Appellants respectfully submit that claim 1 is not anticipated by Martin. Appellants respectfully request withdrawal of the rejection of claim 1 under 35 U.S.C. § 102(a)."

Examiner's Response to Appellent's Arguments

First, The Examiener does not agree the limitations are not taught by Martin. The Examiner believes the best action is for the Board to simple review the reference. The arguments for claim 1 are verbose but reference is very clear. The board is encouraged to look at page 122 and determine if the "Control Condition" an IF structure controls is the control point is active or not. Appellent states the control point would always be TRUE. An object-oriented programmer of ordinary skill would know to reset conditions. In fact, the best response to look over Chapters 9 and 10.

The highlights are :

Chapter 9

Events and Operations - pages 112

Events and State Changes - page 113

The Linking of Operations - page 114

Preconditions and PostConditions - page 115

Clock Events - page 117

Trigger Rules - page 119

Multiple Triggers - page 1120

Control Conditions - page 121

Basic Constructs - page 122 (emphaisis)

Object-Flow Diagrams - page 128

Chapter 10

Rules Linked to Diagram - page 136 (like attach)

Stimulus/Response Rules - page 137

Combining Rules and Chapter nine concepts - page 140 to 142

The Examiner believes the Appellants arguments are evidence of a refusal to acknowledge the most basic concepts of programming. Instead of getting bogged down in resistance the Examiner believes the reference speaks volumes and anticipates the limitations of claim 1. When

viewing the reference with the knowledge of one of ordinary skill the reference anticipates the claim limitaitons of claim 1. The runtime conditions of rules and control points being selected and the paths chosen based on the runtime conditions refered to as "flow control". The Examiner disagrees that the Martin reference does not teach the limitations of claim 1 and that the well known terms are not applicable to Appellent's invention. Appellent's argument's that the Martin reference has a flawed approach to handling control points (active or not) and the selecting of rules is false. Appellent has attempted to distance themself from basic knowledge in the art. The Examiner does not need to provide a verbose response. Examiner disagrees with each of the Appellent's points and feels the Board should be able to see the Martin reference does in fact anticipate the limitations of claim 1.

Disposition for Group I

The Martin reference anticipates all the limitations of claim 1and is proper. The rejection is written to the level of one of ordinary skill in the art at the time of the invention. Appellent is verbose but not persuasive. Rejection is maintained.

B. Group II- Claims 12, 13, 15, 23-28, 35, 46, 51, 52, 59, 59, 72, 74-76, 83, 96 and 98.

Group II claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

C. Group III - Claim 14.

Group III claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

D. Group IV - Claims 31, 55 and 79

Group IV claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

E. Group V - Claims 32, 43, 45, 56, 67, 69, 80, 91 and 93

Group V claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

F. Group VI - Claims 33, 44, 57, 68, 81 and 92

Group VI claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

G. Group VII - Claims 34, 58 and 82

Group VII claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

H. Group VIII - independent claim 16 stands or falls on the limitations of claim 16

Appellant's Arguments

"With regard to claim 16, Martin does not teach defining rules to at least one control point on the basis of an object's class name, method name, and position of the at least one control point in the method. The Examiner alleges that Martin teaches these features for the same reasons as noted with regard to claim 1. Thus, for the same reasons as noted above with regard to claim 1, Appellants respectfully submit that Martin does not actually teach this feature.

In addition, Appellants respectfully submit that claim 1 recites associating a set of rules with each control point based on a class of object, name of method and type of control point. Claim 1 does not recite that the rules are associated based on a noses of a control point in the method. Martin does not teach this feature and the Examiner has failed to show where this feature may be found anywhere in the Martin reference. Thus, the Examiner has not established a case of anticipation with regard to claim 16.

Examiner's Response to Appellants' Arguments Regarding Group VIII

The Examiner responds to the above arguments with regard to claim 16 by stating "Same arguments as above." The Examiner's "arguments above" do not address associating rules based on a position of a control point in a method. Thus, the Examiner has failed to consider all of the features of the claim and has not established anticipation based on the Martin reference with regard to this feature. Moreover, as discussed above, Martin also does not teach the features of associating rules based on an object's class name, method name and position of the control point in a method.

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claim 16 as is required under 35 U.S.C. § 102. At least by virtue of their dependency on claim 16, Martin also does not teach the features recited in dependent claims 17-22. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 16-22 under 35 U.S.C. § 102."

Examiner's Response at Appeal

Appellent's argument that "associating rules based on a position of a control point in a method" is not taught. As seen in claim one Rules are linked to the contructs taught in Chapter 9.

Chapter 10

Rules Linked to Diagram - page 136 (like attach)

Stimulus/Response Rules - page 137

Combining Rules and Chapter nine concepts - page 140 to 142

The rejection is written to one of ordinary skill in the art of object-oriented programming. The need to explain how the contructs of chapters 9 and 10 are implementations in an object-oriented environment is not of ordinary skill. The Martin reference titled "Principles of Object-Oriented Analysis and Design" provides a teaching on building object oriented CASE tools. The Examiner does not understand how the Appellant would not understand what a method is and how the basics of object technology relate to constructs taught in Martin.

Examiner's Disposition on Appeal

Appellent's arguments that fail to understand the implementation of the Martin reference is in an object oriented environment is not persuasive grounds for allowing this group of claims.

I. Group IX - Claims 17 and 22 stands or falls on the limitations of claim 16 and 17.

Appellant's Arguments

"In addition to the above with regard to claim 16, Martin also does not teach many of the specific features set forth in dependent claims 17-22. For example, Martin does not teach activating at least one control point having associated rules, as recited in claim 17. As discussed in depth above, Martin does not even recognize the possibility of having active and nonactive control points, let alone activating a control point. Similarly, this lack of teaching in Martin applies to claim 22 which recites "deactivating the at least one control point." Again Martin does not teach active or nonactive control points and thus, cannot teach deactivating a control point.

Examiner's Response to Appellants' Arguments Regarding Group IX

In response to the above arguments, the Examiner merely responds "See arguments as per above." None of the Examiner's arguments identify where in the Martin reference the activation or deactivation of control points is taught. The Examiner appears to be referring to his erroneous interpretation regarding "flow control" which has been addressed in detail above. Once again, Appellants respectfully submit that there is nothing in the general concept of "flow control" that teaches to activate or deactivate control points. Thus, the Examiner has not shown where the Martin reference identically teaches the features of claims 17 and 22 as is required under 35 U.S.C. § 102."

Examiner's Response and Disposition

Appellent's argument on active and non active control points was part of claim 1 Group I. The Examiner did not find Appellent's arguments persuasive on this topic in Group I. Rejection is maintained.

J. Group X - Claims 18 stands or falls on the limitations of claim 16 and 18.

Appellant's Arguments

"Regarding claim 18, Martin does not teach affecting behavior of the object based on running the rules associated with a control point. As noted above with regard to claim 16, Martin does not teach associating rules with a control point as recited in claim 16. Furthermore, Martin does not teach affecting behavior of an object based on running the rules associated with the control point. The Examiner alleges that this feature is taught on page 381 of Martin but, as with every other allegation made by the Examiner, there simply is nothing on page 381 or any other

page of the voluminous Martin reference that teaches affecting the behavior of an object based on running the rules associated with a control point, the rules being associated with the control point based on an object class, method name, and position in the method.

Examiner's Response to Appellants' Arguments Regarding Group X

The Examiner's response to the arguments regarding claim 18 is to see the argument regarding "active" control points. There is nothing in the Examiner's argument regarding "active" control points that addresses the specific features of claim 18. Thus, the Examiner has failed to show where the Martin reference identically teaches the features of claim 18. Therefore, the Examiner has failed to support his allegation of anticipation of the features of claim 18."

Examiner's Response and Disposition

The Examiner believes this issue was resolved with claim 1 and the linking rules (chapter 10) more than covers the argument above. Rejection maintained.

K. Group XI - Claims 19 stands or falls on the limitations of claim 16 and 19.

Appellant's Arguments

"With regard to claim 19, Martin does not teach that affecting the behavior of the object includes associating different rules to a control point. The Examiner alleges that this feature is anticipated because flow control may be affected by time of day. This has nothing to do with the specific features of claim 19. Nothing in Martin teaches running rules associated with a control point to affect an object wherein the affect is to associate different rules to the control point, as

recited in claim 19. The Examiner simply fails to address this specific feature and rather, rests on generalizations that have nothing to do with the specific feature of the claim.

Examiner's Response n Appellants' Arguments Regarding Group XI

The Examiner's response to the arguments regarding claim 19 is to see the argument regarding "active" control points. There is nothing in the Examiner's argument regarding "active" control points that addresses the specific features of claim 19. Thus, the Examiner has failed to show where the Martin reference identically teaches the features of claim 19. Therefore, the Examiner has failed to support his allegation of anticipation of the features of claim 19."

Examiner's Response and Disposition

The fact that the topics covered in Chapters 9 and 10 are implemented in object technology but Appellent seems to still be arguing this fact was already responded to above. Thsi argument is not presuasive and not deemed grounds for allowance.

L. Group XII - Claims 20 stands or falls on the limitations of claim 18 and 20.

Appellant's Arguments

"Regarding claim 20, Martin does not teach that affecting the behavior of an object includes defining another control point. The Examiner again puts forth irrelevant allegations regarding flow control and then alleges that the features of claim 20 are taught by Martin at page 163. Page 163 is equally irrelevant to the claimed feature as the other allegations regarding Martin made by the Examiner. There is nothing on page 163 or any other page of Martin that teaches running rules associated with a control point to affect the behavior of an object where the affect is

to define another control point. Page 163 merely illustrates an event diagram for registering students. There is nothing in this event diagram that teaches to run rules of control point to affect the behavior of an object such that the affect includes another control point being defined.

Examiner's Response to Appellants' Arguments Regarding Group XI1

The Examiner's response to the arguments regarding claim 20 is to see the argument regarding "active" control points. There is nothing in the Examiner's argument regarding "active" control points that addresses the specific features of claim 20. Thus, the Examiner has failed to show where the Martin reference identically teaches the features of claim 20. Therefore, the Examiner has failed to support his allegation of anticipation of the features of claim 20."

Examiner's Response and Disposition

Appellant's arguments about flow control which is determined by the conditions of the program such as control conditions being true or false and other conditions in the program that affect the behavior has been responded to above. The Appellant stated the Martin reference would always have the control condition of TRUE. The Examiner disagrees. the actual conditions are based on run time conditions not a broad allegation of program constructs in a reference. Rejection maintained.

M. Group XIII - Claims 21 stands or falls on the limitations of claim 18 and 21.

Appellant's Arguments

"Regarding claim 21, Martin does not teach that affecting the behavior of an object includes associating rules to a second control point. The Examiner again alleges that this feature is

taught on page 163 of Martin. Nowhere in the event diagram of page 163 of Martin is there any teaching regarding the running of rules associated with a control point to affect the behavior of the object wherein the affect is to associate rules with a second control point.

Examiner'c Response to Appellants' Arguments Regarding Group XIII

The Examiner's response to the arguments regarding claim 21 is to see the argument regarding "active" control points. There is nothing in the Examiner's argument regarding "active" control points that addresses the specific features of claim 21. Thus, the Examiner has failed to show where the Martin reference identically teaches the features of claim 21. Therefore, the Examiner has failed to support his allegation of anticipation of the features of claim 21."

Examiner's Response and Disposition

The Examiner did not interpret the refernce as offering only one control point to model a solution (write a program). The argument over avtive control points was covered in claim 1.

N. Group XIV - Claims 23 and 27.

Group XIV claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

O. Group XV - Claims 36-39,41,60-63,65,84-87 and 89 stand or fall on the limitations of claim 36

Appellant's Arguments

"With regard to claims 36-39, 41, 60-63, 65, 84-87 and 89, Martin does not teach associating a rule with another method within an object class or associating the rule with another object class. As previously mentioned above, Martin only teaches associating rules with

preconditions, post conditions and control conditions in an event diagram. There is no teaching in Martin that the same rule may be associated with two different methods in an object class or associating the same rule with two different object classes. While similar rules may be associated with different conditions in an event diagram, there is nothing in Martin that teaches that the same rule may be associated with two methods within an object class or two object classes.

The Examiner alleges that this feature is taught on pages 266-268 of Martin. The Examiner alleges that this section of Martin teaches associating rules with more than one object, reuse and inheritance which can all be read on the claimed feature. However, the general teachings of reuse and inheritance do not in themselves teach the specific features recited in claims 36-39, 60-65 and 84-89. There is nothing in Martin that teaches or even suggests to one of ordinary skill in the art that the same rule can be associated with a plurality of methods, as recited in claims 36, 60 and 84 or associated with two different object classes. The Examiner is again engaged in reading into general teachings, the specific teachings found only in Appellants' disclosure without any basis for such reading in of teachings in the reference itself.

Thus, Applicants respectfully submit that Martin does not teach each and every feature recited in claims 36, 60 and 84 as is required under 35 U.S.C. § 102(b). At least by virtue of their dependency on claims 36, 60 and 84, Martin also does not teach the features recited in dependent claims 37-39, 41, 61-63, 65, 85-87 and 89. Accordingly, Applicants respectfully request withdrawal of the rejection of claim 36-39, 41, 60-63, 65, 84-87 and 89 under 35 U.S.C. § 102.

Examiner's Response to AppellantArguments Regarding Group XV

The Examiner responds to the above arguments by stating, with regard to claims 36, 60 and 84, "the ability to call "helper" methods within an object is considered inherent and present in languages such as ANSI standard C++." This has nothing to do with the actual features of claims 36, 60 and 84. Claim 36, 60 and 84 recite associating the rule with another method within the object class. They do not recite "helper" methods and there is no teaching in Martin regarding "helper" methods associating rules with two methods within an object class. The Examiner has failed to show where the Martin reference explicitly teaches this feature and thus, has not met his burden of establishing a case of anticipation based on the Martin reference."

Examiner's Response and Disposition

The Appellent's argument that "the same rule may be associated with two different methods in an object class or associating the same rule with two different object classes" was responded to by stating a term the Examiner expoects one of ordinary skill in object oriented programming to understand the term is a "helper"method. The Examienr did not expect the response to be word matching but concept matching. A "helper" method is a method that resides locally in an object is able to be called from the other methods. This is inherent in object oriented programming. One of ordinary skill in the art would understand ANSI C++ is one of many OO programming languages that support this "helper" methods. Word matching and not concept matching was not deemed persuasive. The Appellent's position the methods of Martin could not fill the role of a "helper" method is not considered inhernet in the art. Rejection is maintained.

P. Group XVI - Claims 42, 47, 66, 71, 90 and 95 stand or fall on the limitations of claim 42.

Appellant's Arguments

"Regarding claims 42, 47, 66, 71, 90 and 95, Martin does not teach the feature of determining if a control point is active. This feature has been address above with regard to claim 1 and thus, claims 42 and 47 define over Martin for similar reasons as noted above with regard to this feature in claim 1. Nothing in Martin teaches that control points may be active or inactive and thus, there is no reason why Martin would teach determining if a control point is active. To the contrary, if a condition exists in the diagram of Martin, it must be active - it either exists or it does not. There is no ability for a condition to exist and it not be active in Martin.

Thus, Appellants respectfully submit that Martin does not teach each and every feature recited in claims 42, 47, 66, 71, 90 and 95 as is required under 35 U.S.C. § 102. At least by virtue of their dependency on claims 42, 47, 66, 71, 90 and 95, respectively, Martin does not teach the features recited in claims 43-44, 67-68, 91-92. Accordingly, Appellants respectfully request withdrawal of the rejection of claims 42-44, 47, 66-68, 71, 90-92 and 95 under 35 U.S.C. § 102.

Examiner's Response to Appellants Arguments Regarding Group XVI

The Examiner does not offer any new rebuttal to Appellants arguments with regard to claims 42-44, 47, 66-68, 71, 90-92 and 95. The Examiner's response is "It appears we have reached a level which dependent claims are relying on prior non persuasive arguments." Thus, the Examiner has not provided any support for the allegation that the features of these claims are anticipated by Martin and merely relies on the erroneous interpretation of Martin previously addressed. Therefore, Appellants respectfully submit that Martin does not teach determining if a

control point is active as discussed above and thus, Martin does not anticipate claims 42-44, 47, 66-68, 71, 90-92 and 95."

Examiner's Response and Disposition

The repeative arguments over active control points were aleady covered in claim 1. Rejction maintained.

Q. Group XVII - Claims 73 and 97.

Group XVII claims are deemed allowable in view of Appellant's argument's on page 16 of Brief.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Todd Ingberg

Primary Examiner

September 17, 2003

Conferees of September 15, 2003

Todd Ingberg	**Kakali Chaki**	**John Chavis**
Primary Examiner	Supervisor Patent Examiner	Patent Examiner

DUKE W YEE

CARSTENS YEE & CAHOON

P.O. BOX 802334

DALLAS, TX. 75380